

Es posible la integración de servicios en la Web con .NET?

Area: Redes y Sistemas Distribuidos

Autores:

C.C. Lía Molinari, lmolinari@info.unlp.edu.ar

Lic. Javier Díaz, javierd@info.unlp.edu.ar

LINTI, Facultad de Informática, Universidad Nacional de La Plata.

Calle 50 y 115, edificio Bosque Oeste, 1er piso. TE:(0221)422-3528, La Plata-Prov. de Bs.As.

Este trabajo trata sobre la integración de servicios sobre la Red, a través de la tecnología Web Services, y, en particular, .NET como alternativa de implementación.

La necesidad de integrar aplicaciones en la Red exige buscar alternativas que provean soluciones interoperables, que funcionen en la diversidad de plataformas, lenguajes y protocolos.

Este nuevo modelo, debe permitir la interacción entre servidores, servidores y clientes, y clientes entre sí. Es por ello que se debe analizar desde varios enfoques: desde las plataformas, desde los lenguajes, como el desarrollador puede participar en la postergada promesa de la reusabilidad del software con abstracción del lenguaje, como integrar servicios sobre la Web con las características mencionadas.

La investigación sobre estos conceptos parte de la bibliografía que se anexa a este artículo.

Estructura del trabajo

En el artículo citado al pie¹, Ethan Cerami, autor de *Web Services Essentials*, se basa en 10 preguntas para guiar al lector en la comprensión del concepto de Web Services y links relacionados.

Esas preguntas son las siguientes:

1. *What is a Web service?*
2. *What is new about Web services?*
3. *I keep reading about Web services, but I have never actually seen one. Can you show me a real Web service in action?*
4. *What is the Web service protocol stack?*
5. *What is XML-RPC?*
6. *What is SOAP?*
7. *What is WSDL?*
8. *What is UDDI?*
9. *How do I get started with Web Services?*
10. *Does the W3C support any Web service standards?*

Estas 10 preguntas, acertadamente escogidas, son una guía para analizar los conceptos relacionados con el tema. En este trabajo, parto de ellas para desarrollar los conceptos asociados con la integración de servicios, en la primer parte del trabajo, y para analizar, en la segunda parte,

¹ Top Ten FAQs for Web Services, By Ethan Cerami.

<http://webservices.xml.com/pub/a/ws/2002/02/12/webservicefaqs.html>

si .NET ofrece la arquitectura y funcionalidad para implementar la integración de servicios a través de la Red.

Web Services ha cambiado no sólo la forma de concebir el desarrollo de aplicaciones, sino las aplicaciones en sí. Para poder entrar en el mundo de los Web Services se hace necesario aclarar, redefinir y definir algunos conceptos. Gran parte de este trabajo trata sobre la introducción a los Web Services.

Palabras claves citadas en la primer parte del trabajo: heterogeneidad, interoperabilidad, extensibilidad, escalabilidad, tratamiento de fallos (detección de fallos, enmascaramiento, tolerancia, recuperación y redundancia), transparencia, cliente, servidor y servicios remotos, el modelo distribuido, el modelo P2P (peer to peer), middleware, ubiquitous connectivity, pervasive computing, aplicaciones: Net Centric, web based, web enabled, three-tier. Business logic, componentes distribuidas, shared context.

La segunda parte del trabajo describe el modelo .NET de Microsoft y lo analiza para la implementación de Web Services.

Web Services

Conceptos y arquitectura

El trabajo se inicia con un análisis rápido sobre la evolución desde la arquitectura de los sistemas de cómputo hasta llegar al modelo distribuido.

Se analiza el cambio en el modelo de accesibilidad y la forma de desarrollar las aplicaciones, comenzando a introducir el concepto de interoperabilidad y la tendencia orientada al modelo distribuido sobre la red.

Con el objetivo de desarrollar aplicación distribuidas altamente integradas que interactúen por XML² entre múltiples servicios WEB, se le dá un lugar relevante a XML y se involucra el concepto de business logic (ver Bibliografía, Sobre business logic) para demostrar una completa integración de los servicios a nivel de todas las capas.

En la bibliografía se citan las referencias a los artículos consultados (Sobre Web Services).

Protocolos

En las preguntas de Ethan Cerami, se hace referencia a los protocolos standards más citados en la filosofía Web Services, como XML, WSDL³, UDDI⁴, SOAP⁵.

No obstante, hay definiciones, entre ellas la avalada por W3C en el Working Draft que se cita al pie⁶, donde *“Web Service es un sistema de software identificado por una URI, cuyas interfaces y enlaces (bindings) son definidos y descriptos usando XML. Este sistema puede ser “descubierto” y usado por otros sistemas de software. Esa interacción debe darse según la manera descripta en la definición, usando mensajes de acuerdo a protocolos de Internet”*.

² XML, eXtensible Markup Language

³ WSDL, Web Service Description Language

⁴ UDDI, Universal Discovery Description & Integration

⁵ SOAP, Simple Object Access Protocol

⁶ Web Services Architecture. W3C Working Draft, 14 Novembre 2002. Editors: Champion (Software AG), Ferris (IBM), Newcomer (Iona), Orchard (BEA Systems)

En esta definición no se nombra SOAP ni WSDL. No obstante, WSA (Web Service Architecture, de W3C) acuerda lo conveniente de basarse en una base común para messaging y descripción (lo define como “una necesidad práctica”).

En la bibliografía se citan las referencias a los artículos consultados (Sobre Protocolos).

El Modelo .NET

.NET es un conjunto de tecnologías de Microsoft. Podemos pensarlo como un sello que Microsoft puso a un conjunto de distintas tecnologías.

Sobre esta tecnología se enuncia que pueden construirse aplicaciones y servicios que funcionan independientemente de la plataforma. Permite la comunicación entre aplicaciones, desarrolladas en distintos lenguajes, en diversas plataformas, integrando distintos dispositivos cliente. Algunos autores, incluso, comparan a .NET como una forma de sistema operativos distribuido a través de Internet, con un soporte optimizado a aplicaciones y servicios.

A través de las experiencia de los usuarios, se ve la necesidad de integrar datos y las preferencias del usuario en una sola aplicación. A través de XML, HTTP y SOAP se logra una solución integrada.

En la bibliografía se citan las referencias a los artículos consultados (Sobre .NET).

Arquitectura .NET

Considerando la arquitectura de los Web services como identificación de sus componentes y la interrelación entre ellas, se describen los componentes de .NET y como se relacionan para lograr la integración de servicios que se quiere obtener.

.NET está compuesto por:

- Un framework
- Visual Studio .NET
- .NET My Services
- .NET Enterprises Services

En la Arquitectura de los Web services, se definen roles que cumplen agentes de software(client service, provider service, discovery agency) y operaciones (find, publish, interact). El artículo se involucra con la arquitectura de .NET con el objetivo de asimilarla a la arquitectura necesaria para la implementación de Web Services.

El framework

El artículo analiza el .NET framework como infraestructura donde los servicios WEB puedan brindarse dentro de las pautas que estuvimos analizando. De esta manera, los programadores pueden enfocarse en la lógica de negocio, más que en la infraestructura de la programación.

Se describe la constitución del framework (CRL y class library). Algunos autores citan ASP.NET como la tercer componente del framework; otros, lo incluyen al hablar de la class library.

Este análisis se orienta a establecer si se cumplen las pautas de interoperabilidad esperada en la filosofía Web Service.

CRL (Common Language Routine)

El análisis de la CRL (Common Language routine), por otra parte, está enfocada en verificar si realmente el programador logra un nivel de abstracción como el enuncao con respecto a manejo de memoria y threads. Esta forma de trabajar establece un cambio por ejemplo, en la forma de enseñar Sistema Operativos, area de mi interés en la docencia.

A través de CRL, se puede indagar la característica language free anunciada.

Framework class library

En cuanto a la librería de clases (.NET framework class library), considerando que sirve para la implementación standard de servicios que se basan en aplicaciones CRL, se analizará en cuanto a su utilidad en la implementación de Web services.

Para implementar esta capacidad multilenguaje, el código fuente es llevado a un código intermedio MSIL (Microsoft Intermediate Language), también llamado IL. En el trabajo se analiza este código, junto con los conceptos de JIT compiler (just in time compiler).

Se realizan también comparaciones de .NET con J2EE.

La integración de servicios

El uso de componentes distribuidas es otro ingrediente en el objetivo Web Services.

Los Building Blocks Services, también llamados Foundation services, son componentes basados en Internet que ofrecen funcionalidad de una forma reutilizable y programable.

Estas componentes a las que puede recurrir el desarrollador para “armar” su aplicación, pueden combinarse con otras componentes. Ejemplo de estos building blocks services, es ***Microsoft Passport***.

.NET utiliza WSDL y XML para la implementación de Web services.

.NET My Services

.NET My Services es un conjunto de web services provistos por Microsoft, accesibles por SOAP. Todos estos servicios, pueden accederse por Internet tanto por parte de usuarios como de aplicaciones.

Incluye un servicio de autenticación, que asegura que sólo los usuarios autorizados pueden acceder a la información que contiene .NET My services (calendarios, inbox, almacenamiento de documentos, profile, etc).

Bibliografía

Sobre Web services

- Web Services Description Language (WSDL) explained. By Carlos C. Tapang. Infotects.
- Creating a Simple XML Web Service. By: Doug Seven
- Why UDDI will succeed, quietly: Two factors Push Web Service Forward. An Analysis Memo from The Stencil Group. 2001
- Web Services Architecture. W3C Working Draft, 14 November 2002. Editors: Champion (Software AG), Ferris (IBM), Newcomer (Iona), Orchard (BEA Systems)
- Understanding SOAP. By Scribner and Stiver

Sobre las aplicaciones

- What is Net Centric Computing?. By Bernard Cole
- The Era of the Net- Centric Computing. By Scott Tilley
- What Are Web-Enabled Applications? by Kathleen s. Carr

Sobre otros protocolos

- COM+ and the Battle for the Middle Tier – Roger Sessions- Editorial Wiley – Año 2000.
- Taller Práctico de CORBA en GNOME- By: Rodrigo Moya. Congreso HispaLinux

Sobre .NET

- Microsoft .NET: Components and Architecture
- Understanding .NET: a tutorial and Analysis. By David Chappell. Editorial Addison-Wesley.
- J2EE vs. Microsoft .NET. A comparison of building XML-based Web Services. By Chad Vawter and Ed Roman (SUN). June 2001.
- J2EE vs. Microsoft .NET Application Server and Web Services Benchmark. Middleware Company. October 2002.
- Introducing Microsoft .NET. By David S. Platt. Editorial: Microsoft Press
- Understanding XML Web Services and the .NET Framework. By: Doug Seven

Sobre Sistemas Operativos y el modelo distribuido

- Operating Systems. William Stallings. Prentice Hall.
- Open Systems. Gary Nutt. Prentice Hall Series.
- Operating Systems Concepts. Silberschatz, Galvin. Editorial: Addison-Wesley

Sobre Business Logic

- Business Process Execution Language for Web Services. Version 1.0. Authors: Curbara, Golan, Kein, Leymann, Roller, Thatte, Weerawarana (IBM, BEA, Microsoft). July 2002
- Choosing the right messaging foundation for business application connectivity. By Scott Garvey. Director Web Services, Microsoft Corporation